

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-267869  
 (43)Date of publication of application : 29.09.2000

(51)Int.Cl.

G06F 9/46  
 H04M 3/00  
 H04Q 3/545

(21)Application number : 11-076703

(22)Date of filing : 19.03.1999

(71)Applicant : NTT COMMUNICATIONWARE CORP

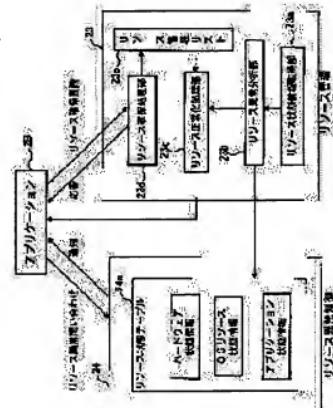
(72)Inventor : SAWAMURA TOSHIHITO  
 TAMURA ZENSHI

## (54) METHOD AND DEVICE FOR CONTROLLING RESOURCE ABNORMALITY, COMPUTER SYSTEM AND RECORDING MEDIUM

## (57)Abstract:

PROBLEM TO BE SOLVED: To obtain a resource abnormality controlling method which integrates processing about resource management and abnormality control in software installed on a computer system and to obtain its resource abnormality controller, a computer system and a recording medium.

SOLUTION: A resource management program 23 and a resource abnormality control program 24 are provided, a resource securing and processing part 23d secures resources in response to a request for resource securing from each application, and a resource abnormality controlling part 24 offers information stored in a resource state table 24a to an application side for an inquiry about resource abnormality from each application. Thus, in the case of resource abnormality, the processing can be shifted to abnormal processing by inquiring of the resource abnormality control program only once at the start of processing, and a part where decision about resource abnormality is made does not have to be included any more in the development of each application.



## LEGAL STATUS

[Date of request for examination] 23.03.2001

[Date of sending the examiner's decision of rejection] 26.10.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]



## 【特許請求の範囲】

【請求項1】 コンピュータシステムにおいて使用される種々のリソースの異常に対する制御を行うリソース異常制御方法において、リソースに対する異常が発生していると判定された場合、その旨を記憶するリソース異常記憶過程と、

前記いずれか複数のリソースに対する複数の識別情報を伴って、該リソースの異常状態を問い合わせる過程と、この問い合わせに応答して、前記リソース異常記憶過程で記憶され、かつ、前記識別情報に対応する情報を提供するリソース異常情報提供過程とを有することを特徴とするリソース異常制御方法。

【請求項2】 請求項1に記載のリソース異常制御方法において、

前記リソースの状態を示すリソース情報を取得する過程と、

取得したリソース情報を分析して、前記リソースに対する異常が発生しているか否かを判定するリソース情報分析過程とを有することを特徴とするリソース異常制御方法。

【請求項3】 請求項2に記載のリソース異常制御方法において、

所定の処理を実行するアプリケーションからの要請に基づいてリソースが確保された場合、該アプリケーションによって実行されるプロセスに対する識別子と、確保されたリソースの種別とを対応付けて記憶するリソース管理過程と、

前記リソース情報分析過程でリソースに対する異常が発生していると判定された場合、前記リソース管理過程において記憶された情報に基づいて該リソースを使用しているプロセスを抽出し、抽出されたプロセスに対して該リソースの異常を回復するための指示を与えるリソース正常化処理過程とを有することを特徴とするリソース異常制御方法。

【請求項4】 請求項3に記載のリソース異常制御方法において、

前記リソース正常化処理過程では、リソース異常と判定されたリソースを使用しているプロセスに対して初期化の指示を与えることを特徴とするリソース異常制御方法。

【請求項5】 請求項1～4のいずれかに記載のリソース異常制御方法を実行することを特徴とするリソース異常制御装置。

【請求項6】 請求項5に記載のリソース異常制御装置を含むことを特徴とするコンピュータシステム。

【請求項7】 請求項1～4のいずれかに記載のリソース異常制御方法を実行するプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、コンピュータシステムにおいて使用されるリソースに異常が発生した場合、このリソース異常に対する制御を行うリソース異常制御方法、リソース異常制御装置、コンピュータシステムおよび記憶媒体に関する。

【0002】

【従来の技術】 図6は、コンピュータシステムに搭載されるソフトウェアの一例として、従来の交換機ソフトウェアの構成を示したものである。同図に示すように、交換機ソフトウェアは、システム全体を統括・制御する基本プログラムであるOSIAと、OSA以外のソフトウェアであって交換処理に纏わる種々の動作を機能分担して実行するアプリケーション2Aとに大別される。アプリケーション2Aは、交換機において通信処理時の実動作を制御する呼処理プログラム20Aと、呼処理を行つための事前データの設定、解除、表示、通信に関する状態情報を表示する保守運用プログラム21Aと、交換機のシステム状態を制御するシステム管理プログラム22Aとから構成される。

【0003】 これらのアプリケーションプログラムは、各処理の実行開始時に必要なリソース（一般に、交換システムでは、トランク、トランザクション等、呼処理で特定の呼や接続に占有されることなく共通に使用される装置、メモリ類を総称してリソースと呼ぶ）を確保して、所定の処理を実行する。このリソースの確保およびリソース異常が発生しているか否かの判定、リソース異常時の対応処理（異常処理といふ）等、リソースに関する処理は、各アプリケーションプログラム毎に別個に行われるとともに、各プログラムの至る箇所で行われていた。

【0004】 このような状況を図7に示す具体例で説明する。同図は、利用者が電話をかける際に受話器を上げることによって、発呼信号が送信され、ダイヤルトーン（発信音）が受話器から聞こえるまでの呼処理プログラム20Aの動作シケンスを示すフローチャートである。まず、利用者が受話器を上げる（S100）と、割り込みが発生する（S101）。そして、タスクが生成されているかどうかが判断される（S102）。例えば、交換機の収容人数（同時に電話交換を受け付けることができる利用者数）が10000人であるものとする。本利用者が10001人目の利用者である場合、リソース不足となりタスクを生成することができず（S102: NO）、異常処理に移行する（S103）。一方、タスクが生成されている場合（S102: YES）には、そのタスクが取得される（S104）。

【0005】 次に、交換機のCPUに対する報告（CPU幅狭という）が発生しているか否かが判断される（S105）。CPU幅狭が発生していると判断される（S105: YES）と異常処理に移行する（S106）。

50 一方、CPU幅狭が発生していないと判断される（S106: NO）と、CPU幅狭が発生していないと判断される（S106: NO）と、

05: NO) と、ダイヤルトーンを利用者側に送るためのハードウェア資源 (DT ハードという) に空きがあるか否かが判断される (S107)。DT ハードに空きがないと判断される (S107: NO) と、異常処理に移行する (S108)。一方、DT ハードに空きがあると判断される (S107: YES) と、リソース管理部 2 3において DT ハードが確保 (リソースが設定) される (S109)。

[0006] 次に、通話路スイッチ (SW ハードという) に故障が発生しているか否かが判断される (S110)。このハート故障が検出されると (S110: YES) 、異常処理に移行する (S111)。ハート故障が検出されなければ (S110: NO) 、SW ハードへのバスが設定されることでリソースが確保され (S112) 、一連のタスクが終了する (S113)。

[0007]

【発明が解決しようとする課題】このように、交換機のアプリケーションでは、通常、各処理を実行していく過程においてリソースが必要となった時点で当該リソースが確保される。そして、上述のアプリケーションの分類においては、各アプリケーションで使用されるリソースを管理する機能が独立して設けられる場合もあったが、リソース異常に対応するためのリソース管理は行われておらず、また、各アプリケーション間で連携したリソースの管理も実行されていなかった。

[0008] したがって、交換機ソフトウェアを開発する際には、次のような問題点があった。まず、各アプリケーションにおいてリソースを確保する処理毎にリソース異常時の対応処理を組み込んでいく形態が採用されていたため、異常処理に関するモジュールの開発量が膨大なものとなっていた。一般に、交換機ソフトウェアの開発においては、まず、正常処理ロジックについて検討し、設計を進める方法が採用される。一方、異常ルートの検討は、この正常なロジックの検討がほぼ終了してから開始される。そして、この正常なロジックが決まった時点で開発スタッフを増員して詳細な検討に入り、異常ルートの検討は開発するグループ内で閉じて行われる。したがって、異常な事象が起る可能性がある場合、それぞれのルートに異常処理を組みしていく開発スタイルとなっていた。

[0009] 通常、交換機ソフトウェアではメインルートに比べて、このメインルートから分岐する分岐ルートの数は莫大なものとなる。したがって、この分岐ルートに異常処理に関するモジュールを盛り込む場合、ソースプログラムのサイズが飛躍的に増大してコーディング量が増えるとともに、バグの潜む可能性が高くなり、しかもそのバグの発見が大変である。このため、全体として開発工数が嵩む結果となっていた。また、コーディングされたプログラムは別途実機レベルでの試験を実施する必要があり、この試験では特異な状況や故障の場合でも

試験対象とされ、試験項目数および試験稼働の増大につながっていた。

[0010] そこで、本発明はこのような課題に着目してなされたもので、コンピュータシステムにおいて使用されるリソースの管理および異常制御に関する処理を集約化したリソース異常制御方法、リソース異常制御装置、コンピュータシステムおよび記録媒体を提供することを目的とするものである。

[0011]

10 【課題を解決するための手段】上記目的を達成するため、第1発明のリソース異常制御方法は、コンピュータシステムにおいて使用される種々のリソースの異常に対する制御を行なうリソース異常制御方法において、リソースに対する異常が発生していると判定された場合、その旨を記憶するリソース異常記憶過程と、前記いすれか複数のリソースに対する複数の識別情報を伴って、該リソースの異常状態を問い合わせる過程と、この問い合わせに応答して、前記リソース異常記憶過程で記憶され、かつ、前記識別情報に対応する情報を提供するリソース異常情報提供過程とを有することを特徴とするものである。

[0012] 第2発明のリソース異常制御方法は、上記第1発明のリソース異常制御方法において、前記リソースの状態を示すリソース情報を取得する過程と、取得したリソース情報を分析して、前記リソースに対する異常が発生しているか否かを判定するリソース情報分析過程とを有することを特徴とするものである。

[0013] 第3発明のリソース異常制御方法は、上記第2発明のリソース異常制御方法において、所定の処理30を実行するアプリケーションからの要請に基づいてリソースが確保された場合、該アプリケーションによって実行されるプロセスに対する識別子と、確保されたリソースの種別とを対応付けて記憶するリソース管理過程と、前記リソース情報分析過程でリソースに対する異常が発生していると判定された場合、前記リソース管理過程において記憶された情報に基づいて該リソースを使用しているプロセスを抽出し、抽出されたプロセスに対して該リソースの異常を回避するための指示を与えるリソース正常化処理過程とを有することを特徴とするものである。

[0014] 第4発明のリソース異常制御方法は、上記第3発明のリソース異常制御方法において、前記リソース正常化処理過程では、リソース異常と判定されたリソースを使用しているプロセスに対して初期化の指示を与えることを特徴とするものである。

[0015] 第5発明のリソース異常制御装置は、上記第1～第4発明のいすれかのリソース異常制御方法を実行することを特徴とするものである。第6発明のコンピュータシステムは、上記第5発明のリソース異常制御装置を含むことを特徴とするものである。第7発明の記録

媒体は、上記第1～第4のいずれかに記載のリソース異常制御方法を実行するプログラムを記録したものである。なお、このプログラムをF D、C D-R O M、データメディア、D V D-R A M等の記録媒体に格納して頒布する（ネットワーク経由でこのプログラムを配信したり、ダウンロードする行為も含む）ことによっても本発明を実施することができる。

#### 【0016】

「発明の実施の形態」との発明の好ましい実施の形態について、以下、添付図面を参照しつつ詳細に説明する。

##### 1. 実施形態の構成

###### 1.1. ソフトウェア構成

図1は、本実施形態に係る交換機のソフトウェア構成を示す図である。同図に示すように、本交換機に搭載されるソフトウェアは、既述の図6に示す交換機ソフトウェアと同様、O S 1 Bと、呼処理プログラム2 0 B、保守運用プログラム2 1 B、システム管理プログラム2 2 Bとを含むアプリケーション2 Bとから構成される。

【0017】さらに、アプリケーション2 Bは、交換機内で使用されるリソース情報（ハードウェア設備数、ハードウェアの状態情報、論理情報の設備数、状態情報）の管理を行うリソース管理プログラム2 3と、このリソース管理プログラム2 3、またはO S 1 Bの内部で管理されるリソースの異常発生時および回復時に通知される情報を分析・制御するリソース異常制御プログラム2 4とを備える。

【0018】このように、従来、交換機ソフトウェアにおいて、呼処理プログラム2 0 B、保守運用プログラム2 1 B、システム管理プログラム2 2 Bで個別に管理されていたリソース情報をリソース管理プログラム2 3に括り出され、当該プログラムにリソース情報を管理する機能が集約化されている。そして、このリソース管理プログラム2 3において異常が検出された場合、リソース異常制御プログラム2 4にその旨が通知される。このリソース異常制御プログラム2 4は、後述するように、リソース管理プログラム2 3およびO S 1 Bから通知されたリソース情報を、ハードウェア関連、O S 関連、アプリケーション関連といったように関連するリソース群にまとめて保持する。

###### 【0019】1. 2. ハードウェア構成

図2は、本実施形態に係る交換機の主要部のハードウェア構成を示す図である。同図に示すように、本実施形態の交換機は、システム全体を統括・制御して各プログラムに実行指令を与えるC P U 1 0 0 a、プログラムが展開されるR A M 1 0 0 b、所定の制御プログラム等が格納されたR O M 1 0 0 c、所定のデータおよびプログラムが格納されたH D D（ハードディスク装置）1 0 0 dを含む中央処理系統装置1 0 0 tと、時分割通話路装置や中継器線信号接続装置等を含む通話路装置1 0 1と、加入者線インターフェース装置等を含む加入者系統装置1 0 2

と、サービスランク装置、課金情報受信装置等を含む付帯系装置1 0 3とを備えて構成され、これらの装置がバス1 0 4を介して相互に接続されている。

【0020】上述の各アプリケーションプログラムは、R A M 1 0 0 bに展開され常時記憶・保持されており、C P U 1 0 0 aの指令化基づいて各プログラムに記述された種々のルーチンが実行される。また、本交換機では、リソース異常が検出された場合でも、C P U 1 0 0 aから通話路系装置1 0 1、加入者系統装置1 0 2、付帯系装置1 0 3に向けた指令により、交換動作への障害を最小限に抑える処理が可能なフォールトトレントシステム（fault tolerant system）が採用されている。

###### 【0021】1. 3. リソース管理およびリソース異常制御プログラムのモジュール構成

図3は、リソース管理プログラム2 3、およびリソース異常制御プログラム2 4のモジュール構成を模式化して表現したものである。まず、リソース管理プログラム2 3はそのモジュール構成において、リソース情報を取得してリソース異常分析部2 3 bに供給するリソース状態情報取扱部2 3 aと、与えられたリソース情報に基づきリソース異常が発生しているか否かの分析を行うリソース異常分析部2 3 bと、リソースの異常が検出された場合、そのリソースを正常な状態に戻す処理を行うリソース正常化処理部2 3 cと、アプリケーション2 B（図3においては、図1の場合と異なり、リソース管理プログラム2 3およびリソース異常制御プログラム2 4を除く）からの要請に基づきリソースの確保処理を行い、当該アプリケーションに対してリソースが確保できたかどうかを通知するリソース確保処理部2 3 dと、確保されたリソースの種別とそのリソースを使用するプロセスのIDが記録されたリソース管理リスト2 3 eとを備える。

【0022】一方、リソース異常制御プログラム2 4は、ハードウェア状態情報、O S リソース状態情報、アプリケーション状態情報からなるリソース状態テーブル2 4 aを備える。アプリケーション2 Bは、リソース異常制御プログラム2 4に対してリソース異常にに関する問い合わせを行い、リソース異常にに関する情報がアプリケーションに提供される。ここで、「ハードウェア状態情報」とは、ハードウェアの故障、工事中（交換機に新たな装置を付加する場合等）、スタンバイ（装置の待機中／稼働中）等の状態を示す情報であり、「O S リソースの状態情報」とは、タスク遷移状態、タスクリソースの空き塞がり、負荷のかかり具合等を示す情報であり、「アプリケーションリソースの状態情報」とはリソースの最大限界、リソースの使用中判定を行うための情報である。

###### 【0023】2. 実施形態の動作

統いて、このようなソフトウェアおよびハードウェアが搭載された交換機におけるリソース異常制御に関する動作

作について説明する。

### 2. 1. 動作の概要

まず、呼処理、保守運用、システム管理の各アプリケーションは、処理を実行する最初の過程でリソース異常制御プログラム2.4にアクセスし、自身のアプリケーションの処理を実行する上で必要となるリソースに異常が発生していないかどうかを、必要となるリソースの識別情報を送信して問い合わせる。この結果、リソース異常なしと判定されると、当該アプリケーションの処理を最後まで実行する一方、必要とするリソースの少なくとも1つに異常があると判定されると、処理の終了もしくはリソース異常が回避されるまで処理を中断する。

【0024】また、リソース状態が異常になった場合、リソース管理プログラム2.3はその旨をリソース異常制御プログラム2.4に通知する。これを受けたリソース異常制御プログラム2.4は、リソース異常に関するアプリケーションからの問い合わせを仲介するインターフェース（リソース状態テーブル2.4a）に対してリソース異常を設定する。そして、リソース正常化処理部2.3cは、リソース異常を回避するため、このリソースを使用して他のアプリケーションを強制的に終了させる処理を行なう。例えば、ある特定の呼に対するプロセスを初期化することで、当該プロセスによって使用されているリソースを解放する「1コールイニシャライズ」等の処理が実行される。

### 【0025】2. 2. 具体的な動作

続いて、具体的な動作例について、呼処理プログラム2.0Bの動作シーケンスを示す図4のフローチャートに基づいて説明する。なお、同図は、既述の従来例の動作を示す図7と対比させて表現したものである。まず、利用者が受話器を上げると（S1）、割り込みが発生する（S2）。呼処理プログラム2.0Bは、必要なリソース（タスクの生成、DTハードの設定、SWハードのパス設定）の識別情報を伴って、これらのリソースの確保が可能であるかどうかの問い合わせをリソース異常制御プログラム2.4に対して行う（S3）。リソース異常制御プログラム2.4は、リソース状態テーブル2.4aに記憶された情報を参照して、呼処理プログラム2.0Bの必要とするリソースに異常が発生しているか否かの情報を提供する。

【0026】これを受けた、呼処理プログラム2.0Bは、必要とするリソースのいずれか1つでも異常がある旨の情報が供給された場合（S3: NO）、異常処理に移行する（S4）。一方、必要とするリソースのすべてが正常である旨の通知がされた場合（S3: YES）には、リソース管理プログラム2.3cに対してリソースの確保を要請する。すなわち、まず、タスクを取得して（S5）、DTハードが確保されてリソースが設定される（S6）。さらに、SWハードへのパスが設定されリソースが確保され、タスクが終了する（S8）。

【0027】図5は、上記図4に示すシーケンスにおいて、必要とするリソースが正常であると確認された後、実際にリソースを確保する場面におけるリソース管理プログラム2.3とリソース異常制御プログラム2.4の動作を示したものである。まず、呼処理プログラム2.0BからDTハードリソースの確保が要求される（S10）と、リソース管理プログラム2.3は、DTハードがシステム搭載数の最大に達したか否かを判断して（S11）、DTハードの空きがない場合（S11: YES）には、その旨をリソース異常制御プログラム2.4側に通知することで、リソース状態テーブル2.4aにその旨が記憶される（S12）。一方、最大数に達していない場合（S11: NO）には、呼処理プログラム2.0BにDTハードリソースの確保が完了した旨を通知する（S13）。

### 【0028】3. 実施形態の効果

（1）呼処理、保守運用、システム管理の各アプリケーションは、処理の冒頭で1回だけリソース異常制御プログラムに対して問い合わせを行うことにより、必要とするリソースに異常が発生しているか否かを判断して、リソース異常の場合には異常処理に移行することができる。

（2）したがって、従来のように各アプリケーション毎に任意の箇所でリソース異常の判定を行う場合と比べて処理が一本化され、各アプリケーションの開発において、リソース異常に関する判定を行う部分を盛り込む必要がなくなり、リソース異常に関する部分の検討ポイント数が削減されることと相俟って、全体の開発工数が大幅に削減される。

（3）これに関連して、対象交換機の機種変更やバージョンアップに対する異常処理部の修正個所を最小限にすることができる、ソフトウェア生産工程における作業能率が向上する。

（4）さらに、交換機ソフトウェアの試験を実行するための試験稼働に要する時間も大幅に削減される。

### 【0029】4. 変形例

本発明は、上記実施形態に限定されるものではなく、例えば以下のように種々の変形が可能である。

（1）本実施形態では、リソース管理部でリソース異常分析を行いリソース異常と判定された場合の処理（上記実施形態では、1コールイニシャライズ）を実行する指令を発行することとしたが、この異常分析とリソース正常化処理をリソース異常制御部で行なうようにしてもよい。

（2）本実施形態では、交換機を例に挙げて説明したが、交換機以外のコンピュータシステムに対しても本実施形態に係るリソース異常制御方法を適用することができます。

### 【0030】

【発明の効果】以上詳細に説明したように、この発明に

よれば、コンピュータシステムに搭載されるソフトウェアにおいて、リソース情報を管理してリソースの異常処理を集約化することで、各プログラム毎にリソース管理およびリソース異常に対する処理を盛り込む必要がなくなる。

【図面の簡単な説明】

【図1】本実施形態にかかる交換機ソフトウェアの構造を示す図である。

【図2】本実施形態にかかる交換機の主要部のハードウェア構造を示す図である。

【図3】本実施形態にかかる交換機ソフトウェアにおいて、リソース管理部とリソース異常制御部の構成を中心と表現したものである。

【図4】本実施形態のリソース異常制御方法を用いた場合の具体例を示すフローチャートである。

【図5】本実施形態のリソース異常制御方法を用いた場合のリソース管理部とリソース異常制御部との相互動作を示すフローチャートである。

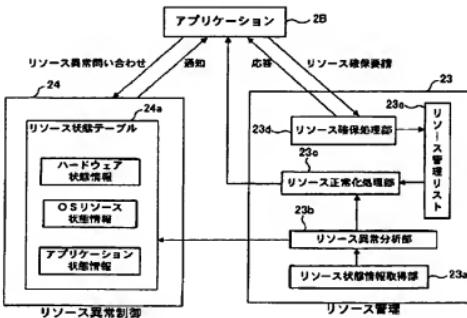
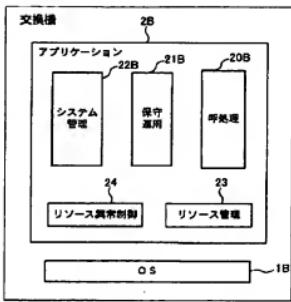
\* 【図6】交換機における従来のソフトウェア構造を示す図である。

【図7】従来の呼処理プログラムにおける異常処理の動作を示すフローチャートである。

【符号の説明】

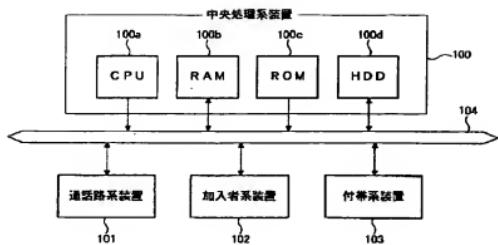
1 A, 1 B	OS
2 A, 2 B	アプリケーション
20 A, 20 B	呼処理プログラム
21 A, 21 B	保守運用プログラム
10 22 A, 22 B	システム管理プログラム
2 3	リソース管理プログラム
2 3 a	リソース状態情報取得部
2 3 b	リソース異常分析部
2 3 c	リソース正常化処理部
2 3 d	リソース確保処理部
2 3 e	リソース管理リスト
2 4	リソース異常制御プログラム
2 4 a	リソース状態テーブル

【図1】

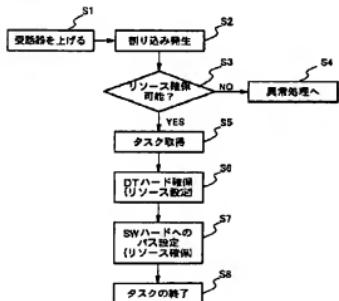


【図3】

【図2】



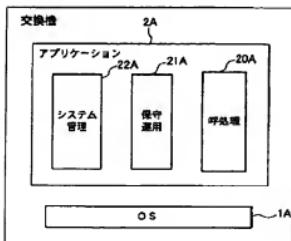
【図4】



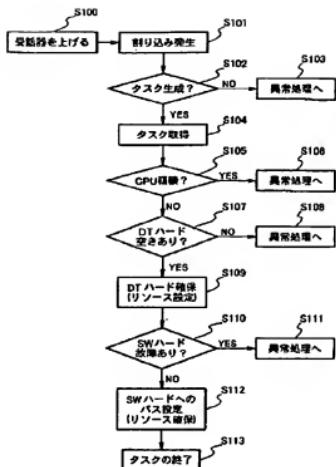
【図5】



【図6】



【図7】



フロントページの続き

(72)発明者 田村 善嗣  
 東京都港区港南一丁目9番1号 エヌ・ティ・ティ・コミュニケーションウェア株式会社内

F ターム(参考) 5B098 AA05 GA02 GA06 GB05 GD02  
 GD03 GD07 GD12 GD14 GD21  
 5K026 AA03 CC07 FF02 GG14  
 5K051 AA10 EE01